1

UNCLASSIFIED

AD-A206 962

DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFATL-TR-89-01

6a. NAME OF PERFORMING ORGANIZATION
Westinghouse Electric
Corporation

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Guidance & Control Branch
Aeromechanics Division

6c. ADDRESS (City, State, and ZIP Code)
Westinghouse Electric Corporation
Development Operations Division
PO Box 1693, Baltimore MD 21203

7b. ADDRESS (City, State, and ZIP Code)
Air Force Armament Laboratory
Eglin AFB FL 32542-5434

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
AD/AFATL

8b. OFFICE SYMBOL (If applicable)
FXG

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F08635-87-C-0107

8c. ADDRESS (City, State, and ZIP Code)
AFATL/FXG
EGLIN AFB FL 32542-5434

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
62602F	2567	01	53

11. TITLE (Include Security Classification)
Guidance Instruction Set Architecture: Missile Computing Power for the 90's (u)

12. PERSONAL AUTHOR(S)
Harto, Debra L. York, Lt George W. P.

13a. TYPE OF REPORT
Tech Paper

13b. TIME COVERED
FROM Sep 87 TO Jan 89

14. DATE OF REPORT (Year, Month, Day)
February 1989

15. PAGE COUNT
8

16. SUPPLEMENTARY NOTATION
N/A

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
09	02	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Missile Computer, Instruction Set Architecture, Computer Hardware, (edc)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

With the ever increasing processing demands placed on weapon systems, current general purpose processors cannot meet the throughput requirements. Also, processors optimized for executing applications implemented in the DOD mandated language, Ada, are necessary. However, a balance between customized hardware and off-the-shelf convenience must be sought. Such a balance is being pursued in the Guidance Set Architecture (GISA) program sponsored by the Air Force Armament Laboratory. In the GISA program, an instruction set architecture optimized for processing tactical missile guidance and control algorithms written in Ada is being developed using off-the-shelf products to the maximum extent possible. This paper provides a description of the program's goals, objective, and accomplishments as well as possible plans for future efforts in this area. Keywords: [illegible]; [illegible] computers; computer architecture;

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
DEBRA L. HARTO

22b. TELEPHONE (Include Area Code)
(904) 882-2961

22c. OFFICE SYMBOL
AFATL/FXG

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

GUIDANCE INSTRUCTION SET ARCHITECTURE: MISSILE COMPUTING POWER FOR THE 90s

Lt George York and Debra Harto

ABSTRACT

Tactical missile guidance and control algorithms have always been computationally intensive. With the ever increasing processing demands placed on weapon systems, current general purpose processors cannot meet the throughput requirements. Also, processors optimized for executing applications implemented in the DOD mandated language, Ada, are necessary. However, a balance between customized hardware and off-the-shelf convenience must be sought. Such a balance is being pursued in the Guidance Set Architecture (GISA) program sponsored by the Air Force Armament Laboratory. In the GISA program, an instruction set architecture optimized for processing tactical missile guidance and control algorithms written in Ada is being developed using off-the-shelf products to the maximum extent possible. This paper provides a description of the program's goals, objectives, and accomplishments as well as possible plans for future efforts in this area.

BACKGROUND

Currently, general purpose microprocessors, similar to those used in personal computers, workstations, and even home appliances, are also used in weapon systems. However, as tactical missile guidance and control algorithms become more complex, neither general purpose nor MIL-STD-1750A microprocessors will meet the processing requirements. Future guided weapons will require greatly improved computational capability to execute modern guidance algorithms, utilize artificial intelligence, and support intensive seeker signal processing.

Current guidance and control microprocessors are not capable of supporting these requirements because general purpose computers sacrifice performance for flexibility. In order to meet high performance requirements, general purpose machines have given way to special purpose instruction set architectures (ISAs) for certain applications such as radar signal processing. An ISA comprises the central processing unit (CPU) instructions and internal characteristics as seen by an assembly language programmer. The ISA is one level lower than that which a high order language (HOL) programmer must consider. However, in real-time applications, the HOL programmer must be concerned with the efficiency of the execution of the HOL code which is directly dependent on the efficiency of the ISA. To date, this special purpose ISA concept has not been applied to missile guidance and control microprocessors.

To meet this challenge, in July 1986 the Air Force Armament Laboratory (AFATL) initiated the Guidance Instruction Set Architecture-Phase 1 (GISA-1) program, a basic research effort recently completed by LTV. The objective of GISA-1 was to determine the feasibility of developing a 32-bit ISA optimized for missile guidance and control. The results of the program were that the concept was feasible and that an optimized ISA would be more efficient than commercial general purpose microprocessors at executing guidance and control applications. However, certain questions were still unanswered: Can an off-the-shelf ISA meet the throughput requirements? Is a hybrid of customized and off-the-shelf hardware feasible?

GISA-2 PROGRAM APPROACH

In September 1987, the Air Force Armament Laboratory initiated the Guidance Instruction Set Architecture-Phase 2 (GISA-2) program. GISA-2 is aimed at developing a 32-bit ISA optimized for processing tactical missile guidance and control algorithms written in Ada. Both customized and off-the-shelf hardware options are being explored. The GISA-2 contract was awarded to Westinghouse Electric Company (WEC).

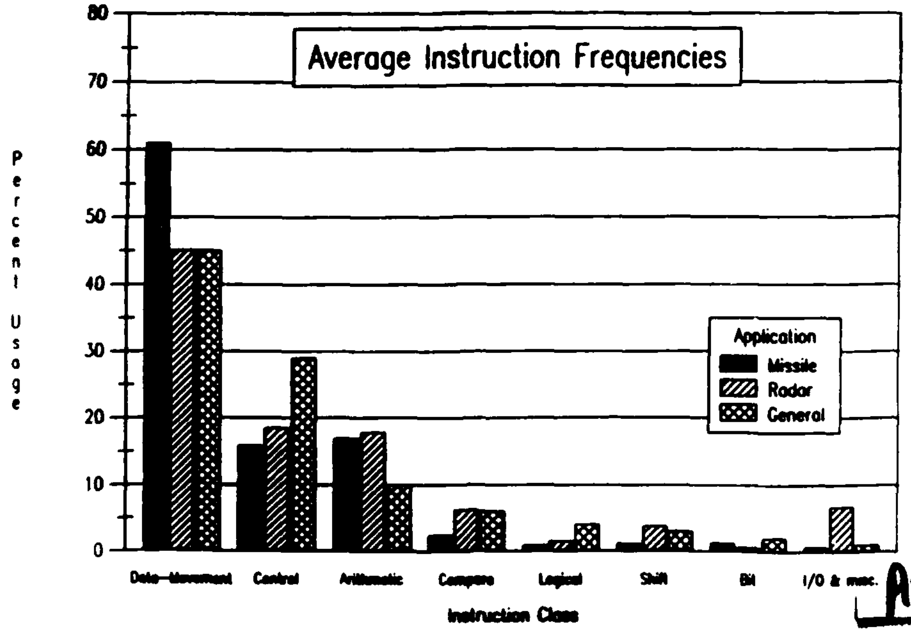
The GISA program is divided into the following six tasks: (1) software domain analysis, (2) instruction set architecture survey, (3) hardware design decisions, (4) hardware design and implementation, (5) Ada tool set development, and (6) evaluation. These tasks are described in the following paragraphs.

Software Domain Analysis

A domain analysis was conducted from September 1987 to April 1988 to determine the processing requirements of guidance algorithms. Missile and radar operational flight software were analyzed to determine the assembly-level instructions used most often in guidance and control algorithms. The software analyzed in the domain analysis included guidance software from the Advanced Medium Range Air-to-Air Missile (AMRAAM), a WEC in-house classified missile, the F-16 front end radar system (APG-68), and an aircraft radar called HELRATS, the High Energy Laser Ranging And Targeting System. Also, the Common Ada Missile Packages (CAMP) guidance and navigation benchmarks were analyzed. The CAMP program developed 454 efficient, reusable Ada missile parts and armonics benchmarks representative of operational missile software. The CAMP benchmarks were analyzed to determined static and dynamic instruction frequency, procedure call depth, and number of parameters passed in subroutine calls.

It was concluded that the instruction frequency of radar and missile software was similar (see figure). Data movement instructions were

Average Instruction Frequencies

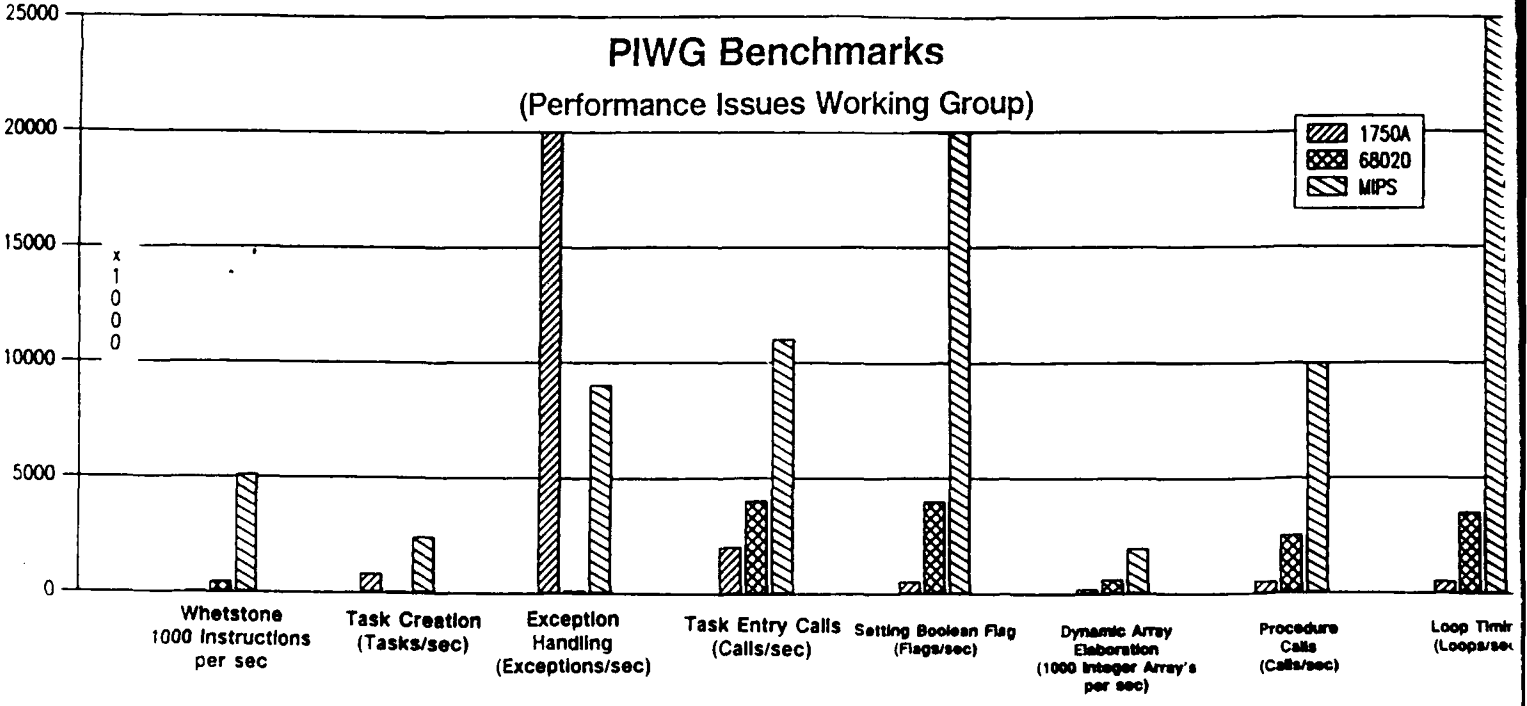


the most frequently used operations and accounted for over 60 percent of the missile assembly-level instructions and 45 percent of the radar code. The second most frequently used instructions were control operations (branch, loop, subroutine call, etc) which accounted for 15 to 20 percent of the code. Math operations accounted for approximately 15 percent of the code. Another conclusion was that predominantly floating point arithmetic rather than fixed point arithmetic was performed. It was also determined that an average of three parameters were passed in subroutine calls, and the average call depth was two subroutines. Since both the average number of parameters passed in subroutine calls and the average call depth were small, it was decided that multiple register sets are unnecessary.

Instruction Set Architecture Survey

After the guidance and control requirements were defined, a survey was conducted to determine if an off-the-shelf processor could meet these requirements. Specific areas of interest were (1) support for Ada tasking, (2) amount of overhead involved in context switching and subroutine calls, and (3) amount of built-in support for multiprocessing and memory management schemes. The ISA survey was conducted by executing the SIGAda Performance Issues Working Group (PIWG) benchmarks on three off-the-shelf microprocessors. The benchmarks used included Whetstone, Task Creation, Exception Handling, Dynamic Array Elaborations, Setting a Boolean Flag, Procedure Calls, Task Entry Calls, and Loop Timing. The three candidate microprocessors were a MIL-STD-1750A, a Motorola 68020, and a MIPS Computer Systems R2000. The MIL-STD-1750A processor used was the Fairchild 9450 chip which was driven by a 25 MHz clock and used an unoptimized ACT Ada compiler. Another candidate was the Motorola 68020 which operated at 16 MHz and used a Verdix Ada compiler (Optimized with Suppression). The third processor was a MIPS Computer Systems R2000 which ran at 16 MHz and used an unoptimized Verdix Ada compiler.

The following figures show the relative performance of the three processors for these benchmarks. The performance of the MIPS processor far exceeded



the other processors in all benchmarks except exception handling. It was concluded that reduced instruction set computer (RISC) architectures, like the MIPS, offer the potential for the highest throughput, and no off-the-shelf processor can meet all the guidance and control requirements without some modification.

Hardware Design Decisions

Based on the ISA survey and the domain analysis results, it was decided that the baseline for the GISA-2 hardware will be the DARPA Core MIPS ISA. MIPS is an acronym for Microprocessor without Interlocked Pipeline Stages. In complex instruction set computer (CISC) architectures, the number of clock cycles per instruction varies. Therefore, interlocked pipeline stages are used to provide control and timing in order to execute instructions efficiently. However, implementing the interlocked pipeline stages is complicated and the extra hardware logic necessary takes up "real estate" on the chip. Since the MIPS architecture does not use interlocked pipeline stages, it is simpler than a CISC architecture and takes up less room on a chip. In order for the MIPS architecture to process instructions efficiently, each instruction should execute in one clock cyle. For those few that do not, such as those that require memory access, the compiler inserts NO-OP (no operation) instructions to insure proper timing.

The DARPA Core MIPS ISA is a government-owned specification for a reduced instruction set computer (RISC) architecture. The document, Core Set of Assembly Language Instructions for MIPS-based Microprocessors, defines the instruction set for MIPS-based microprocessors. The intent was that compiler writers could target their compilers to the MIPS core ISA, and hardware designers could target their hardware to execute the assembled code. The document was written with the purpose of allowing one Ada compiler to work with several MIPS machines without requiring the compiler to be rewritten for each MIPS implementation. The MIPS core ISA document was written to be specific enough to define a complete ISA, yet flexible enough not to confine the hardware designers. However, due to this flexibility, the document is too vague in some areas and some other areas are not addressed at all. For example, the document does not address the issues of multiprocessing, fault tolerance, and operating system support. Also, the register/data formats, interrupt structure, input/output, memory management, and real-time clocks are not well defined. This has led to different interpretations of the MIPS core ISA.

Several processors based on the DARPA Core MIPS ISA have been developed. For example, MIPS Computer Systems has produced a series of processors, the R2000 and R3000, based on their interpretation of the MIPS core ISA document. The GISA breadboard will use the MIPS Computer Systems chip set to execute the GISA ISA. In a later phase, the GISA processor may be constructed on a chip using VHSIC technology.

The GISA-2 ISA will be designed using the DARPA Core MIPS ISA as a baseline, but the designers will add instructions and interpret the ISA as necessary to optimize for guidance and control applications. The following enhancements to the MIPS core ISA are proposed for the GISA-2 architecture:

a. instructions to support extended precision math such as add and subtract with carry which can provide a 50% improvement in the time required to perform double precision arithmetic.

b. instructions for bit operations, such as "test and set bit in memory", which aid in semaphore signaling operations and device control.

c. logical instructions, such as AND and OR, will be used to perform operations such as "clear and set bit."

d. bit field instructions such as "find first bit set/cleared" and "extract/insert bit field" in order to support Ada runtime memory allocation and device control.

e. interfaces to co-processors in order to provide a standard means of accessing unique hardware operations provided by co-processors.

f. user-defined instructions.

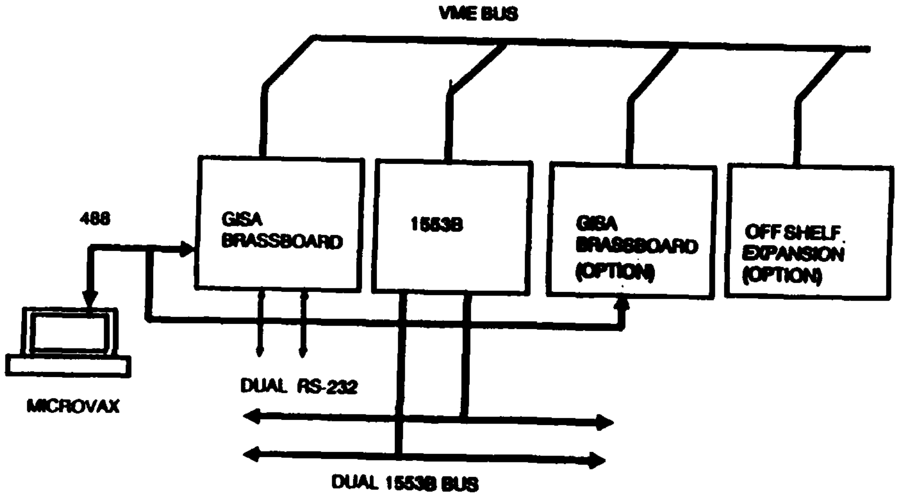
The GISA-2 architecture will not contain instructions to support link list manipulations. Analysis of the guidance and control software showed that link list operations were rarely executed. Many processors provide support for linked lists, but these operations are seldom executed by the runtime executive or the embedded application code. Therefore, since little if any performance improvement can be gained in a RISC processor by implementing link list instructions, the GISA-2 computer will not contain any unique instructions to support this feature.

Exponential and transcendental functions are traditionally found in guidance and radar processing algorithms, but these functions will not be implemented in the GISA hardware. Instead, they will be provided as callable software routines that will reside in the runtime library.

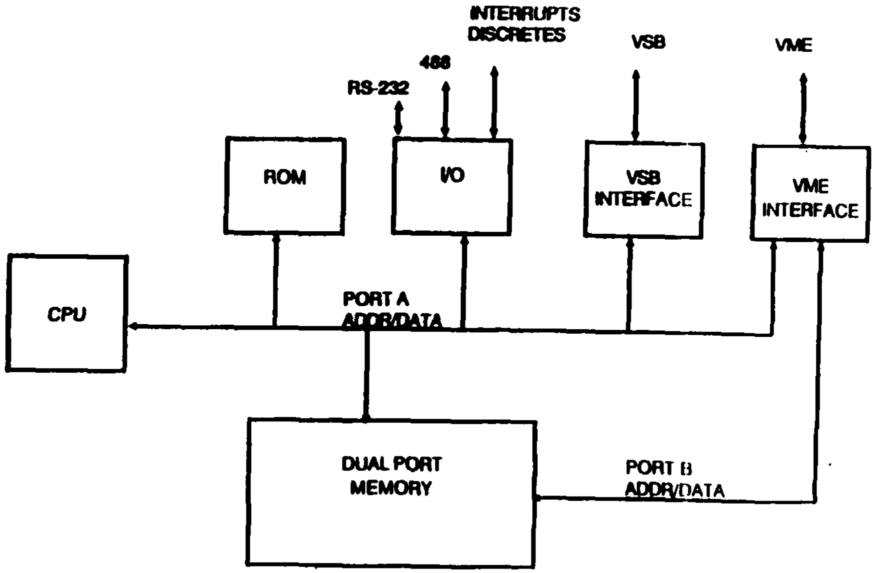
Hardware Design and Implementation

The GISA-2 hardware products are a breadboard CPU which executes the GISA code and an accompanying brassboard computer. The following figures show the breadboard CPU and its interfaces, and a diagram of the GISA brassboard computer. The brassboard supports an IEEE 488 interface, a RS-232 serial interface, a VME/VSB bus, and a dual 1553B bus. The breadboard contains 1 megabyte of RAM, 64k bytes of instruction cache, 64k byte of data cache, and 256k bytes of UVPROM (Ultra-Violet Programmable Read-Only Memory). The UVPROM is used for boot and diagnostic software, as well as the runtime executive. In order to process floating point data quickly, the breadboard CPU will be supported by a floating point coprocessor (double and single precision). The breadboard supports two 32-bit programmable, resetable interval timers, one real-time clock, 32 general purpose interrupts, 32 1-bit discrete ports, 32 32-bit general purpose registers, two multiply/divide registers, one program counter, and 16 64-bit floating point registers.

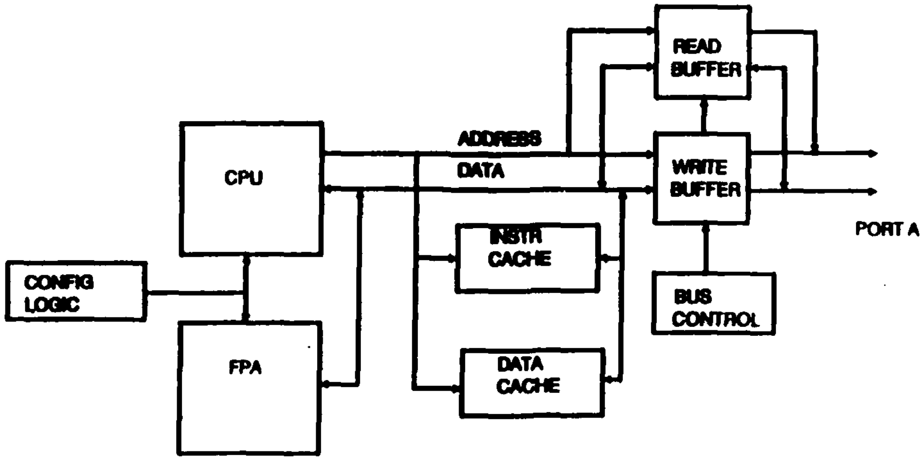
GISA SYSTEM ARCHITECTURE



BRASSBOARD CONFIGURATION

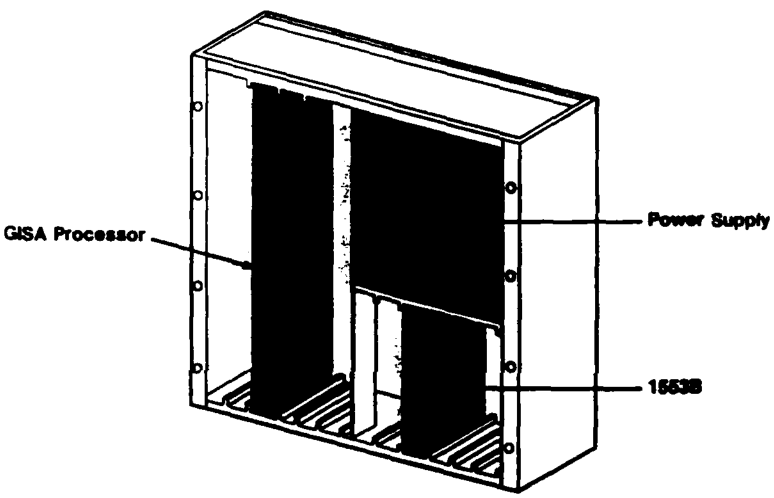


CPU CONFIGURATION

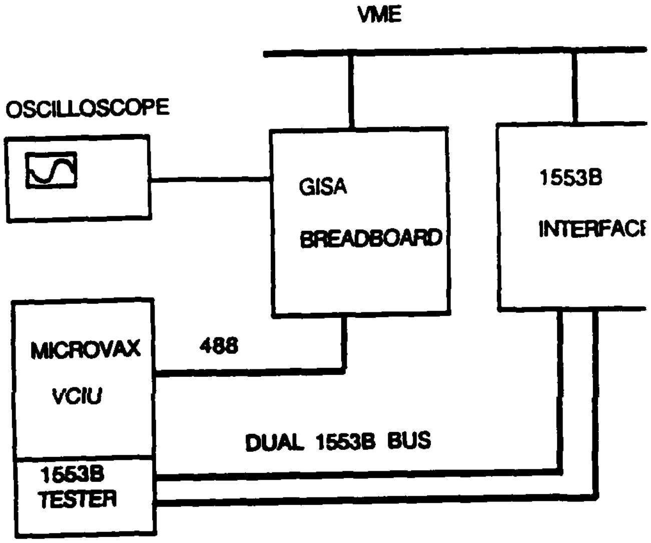


The hardware will be delivered in a rack as shown below. The rack has expansion slots to hold additional GISA processor breadboards. Westinghouse has developed an operating system, Kernal Operating System (KOS), that will be modified to support multiprocessing with numerous GISA brassboards. For testing the GISA computer, benchmark programs will be downloaded from a MicroVAX to the GISA brassboard via a 488 bus. Residing on the MicroVAX is a 1553B tester that can examine the 1553B interface directly. The GISA breadboard may communicate with a device such as an oscilloscope via its discrete ports and with other processors across the VME bus.

System Configuration



Test Configuration



Ada Tool Set Development

Since it is essential to be able to develop applications in Ada on the GISA-2 processor, AFATL is developing an Ada tool set under the GISA-2 effort. InterAct under subcontract to Westinghouse is developing the GISA Ada tool set. The Ada tool set includes an Ada compiler, assembler, symbolic debugger, linker/loader, and runtime executive. The compiler will be hosted on a MicroVax and executable code will be downloaded from the MicroVax to the GISA hardware. InterAct is retargeting an existing Ada compiler to the GISA hardware and is also responsible for the assembler and linker/loader. The symbolic debugger will be a modified version of the VAMP (VHSIC Avionics Modular Processor) debugger. The runtime executive will be a modified version of the Westinghouse KOS. InterACT is tailoring the Ada compiler to interface with the KOS runtime executive which will allow efficient execution of Ada application code.

The quality of the Ada compiler has a significant impact on the target processor's performance. Since InterACT is retargeting a mature Ada compiler to the GISA hardware, it should be more efficient than one developed from scratch. Since the government is not in the business of maintaining software tools, the Air Force will only own a license to use the compiler. InterACT is wholly responsible for maintaining and marketing the compiler.

Evaluation

The final task of the GISA-2 program is to evaluate the performance of the GISA hardware versus a Motorola 68020, a MIL-STD-1750A, and at least one other 32-bit processor. The third processor is yet to be specified. The AFATL-developed CAMP benchmarks will be executed on each of the computers to collect performance data. The data will be studied to determine how the performance of the GISA-2 hardware compares to that of the off-the-shelf processors.

PRODUCTS

The GISA-2 deliverables include the GISA breadboard and brassboard along with a specification for manufacturing the GISA-2 hardware into a VHSIC chip. This chip may be constructed in a follow-on project if the performance results of GISA-2 are positive. The government will receive a license to use the GISA Ada tool set. The government will own the GISA debugger, but will be licensed to use the Ada compiler. A final report documenting the development and evaluation of the GISA-2 hardware will be delivered. The hardware will be integrated during May-Sept 89 and the evaluation will be conducted during Sept-Dec 89.

FUTURE DIRECTIONS

If the GISA-2 processor performs as well as expected, a GISA-3 project may be undertaken. In GISA-3, a VHSIC chip will be manufactured following the specification delivered as part of the GISA-2 project, and the brassboard computer will be form-factored for a weapon system. Real-time simulations will be performed with the GISA-3 computer as a processor-in-the-loop. Advanced guidance laws which are computationally intensive will be analyzed on this system. Also, experiments in multiprocessing will be performed.

CONCLUSION

The GISA program will result in an instruction set architecture optimized for guidance and control algorithms written in Ada. To demonstrate and evaluate the ISA, a GISA processor and Ada software tools will be developed. The performance comparison of the GISA processor (a custom/off-the-shelf hybrid) versus general purpose processors will show if significant gains can be made in processing guidance and control algorithms due to an optimized ISA. If the GISA processor performs as expected, it will allow researchers to more efficiently process modern guidance and control algorithms in the real-time environment in support of future tactical weapons.